Progressively Adding Objectives: A Case Study in Anomaly Detection

Luis Martí, Arsene Fansi-Tchango, Laurent Navarro, Marc Schoenauer

Progressively Adding Objectives: A Case Study in Anomaly Detection

Luis Martí
TAU, INRIA, LRI/CNRS/UP-Sud
1 rue Honoré d'Estienne d'Orves
Palaiseau, France 91120
Universidade Federal Fluminense
Av. Gal. Milton Tavares de Souza s/n
Niterói, RJ, Brazil 24210-346

Arsene Fansi-Tchango
Laurent Navarro
ThereSIS, Thales Group
1 avenue Augustin Fresnel
Palaiseau, France 91767

Marc Schoenauer
TAU, INRIA, LRI/CNRS/UP-Sud
1 rue Honoré d'Estienne d'Orves
Palaiseau, France 91120

ABSTRACT

One of the principles of evolutionary multi-objective optimization is the conjoint optimization of the objective functions. However, in some cases, some of the objectives are easier to attain than others. This causes the population to lose diversity at a high rate and stagnate in early stages of the evolution. This paper presents the progressive addition of objectives (PAO) heuristic. PAO gradually adds objectives to a given problem relying on a perceived measure of complexity. This diversity loss phenomenon caused by the nature of a given objective has been observed when applying the Voronoi diagram-based evolutionary algorithm (VorEAl) in anomaly detection problems. Consequently, PAO has been first directed to address that issue. The experimental studies carried out show that the PAO heuristic manages to yield better results than the direct use of VorEAl on a group of test problems.

CCS CONCEPTS

•**Security and privacy** → *Intrusion detection systems;* •**Computing methodologies** → Anomaly detection; **Genetic algorithms;**

KEYWORDS

Evolutionary multi-objective optimization, many-objective optimization, anomaly detection

ACM Reference format:
Luis Martí, Arsene Fansi-Tchango, Laurent Navarro, and Marc Schoenauer. 2017. Progressively Adding Objectives:
A Case Study in Anomaly Detection. In *Proceedings of GECCO '17, Berlin, Germany, July 15-19, 2017,* 8 pages.
DOI: http://dx.doi.org/10.1145/3071178.3071333

1 INTRODUCTION

Many practical problems can be posed as multi-objective optimization problems (MOPs) [9]. MOP solutions call for the conjoint optimization of a set of possibly contradictory objective functions. In the general case, the solution to a MOP is a set of equally good trade-off solutions that is known as the Pareto-optimal set. Evolutionary multi-objective optimization algorithms (EMOAs) [9] have been found to be a competent approach to dealing with MOPs.

One particularly interesting class of MOP appears as the number of objective functions in $\boldsymbol{F}$ grows. When this happens, the corresponding MOP becomes more challenging to the corresponding optimizer. Because of this, these many-objective problems [18] are of particular interest in the area.

In general EMOA practice all objectives are considered to be equally important. Therefore, the same computational effort is dedicated to each when carrying out the optimization process. This makes sense in theoretical or experimentation contexts, where it is not important to take into account any temporal or spatial computational resource restrictions. However, in real-world practice, it is necessary to focus available resources on areas of the Pareto-optimal set of practical value. This has prompted the emergence of preference handling in EMOAs as one of the main research areas of the field [13, 24, 25].

One successful application of an evolutionary many-objective approach is the one put forward by the Voronoi Diagram Evolutionary Algorithm (VorEAl) [20]. VorEAl evolves Voronoi diagrams that are used to classify data in an anomaly detection context [8]. Anomaly detection can be defined as a semi-supervised classification problem where: (i) it is necessary to correctly identify 'anomalous' from 'normal' instances present in the learning dataset and (ii) to be able to detect anomalous data that are not known beforehand. Because of this, posing the anomaly detection problem as a MOP involves two main classes of objective functions. On the one hand, some objectives, like classification accuracy and recall, are meant to quantify the quality of the model as a classifier. On the other hand, other objectives are meant to produce a compact representation of 'normal' data that can be used to detect anomalies of the second kind (akin outlier detection).

One of the drawbacks of the original formulation of VorEAl is that the size (number of Voronoi cells) of the Voronoi diagrams must take values in an *a priori* given interval. This is an important inconvenience as, when dealing with real-world problems, it is impossible to estimate the suitable limits of that interval. In an attempt to improve the original VorEAl, we experimented with adding an additional objective to VorEAl meant to minimize the size of the diagrams and, therefore, avoid the need of expressing the aforementioned interval in an explicit form.

This modification yielded negative but thought-provoking results. Analyzing the phenomenon in depth, it was possible to establish

GECCO '17, Berlin, Germany

DOI: http://dx.doi.org/10.1145/3071178.3071333

that populations evolved rapidly towards the minimization of the new objective, thus losing diversity at a high rate. This can be explained by the fact that it is rather simple to optimize the diagram size minimization objective –as it implies just creating smaller individuals– while it is more complicated to evolve diagrams that improve objectives like accuracy or recall. Furthermore, small individuals would have, by definition, a low representation capacity and, hence, would not be able to provide an adequate representation of the learning data. Therefore, this new objective not only degraded the diversity of the population but made it virtually impossible to attain the primary and more important objectives.

In ideal conditions, it can be hypothesized that applying a proper selection method should guarantee that no locally Pareto-optimal individuals would be lost. However, even if their impact on the optimization process might be marginal, if VorEAl is left to run for a sufficient number of iterations, the diversity loss issue eventually could be overcome. However, this alternative is not viable in practical conditions where solutions must be obtained in a reasonable time budget.

Furthermore, even though, to the best of our knowledge, this issue has not been previously dealt with, similarly conditioned problems emerge in other areas of evolutionary computation. That is the case, for example, of genetic programming [17] where some attempts have been made to reducing bloat using EMOA-based approaches (cf. [5, 10]) leading to problems with similar characteristics. In this case, smaller programs are much more simple to produce than programs that meet the objectives of interest.

Summarizing the previous discussion, we can state that in many –if not all– real-life MOPs, there is a group of primary objectives that represent the features that must be improved in the artifacts being optimized; while a set of secondary objectives are intended to capture some desires or preferences regarding these artifacts. In some cases, these secondary objectives are much easier to attain. This leads to a loss of population diversity and hinders the achievement of adequate solutions with respect to the primary objectives. That is, paraphrasing Orwell's Animal Farm [22], *all objectives are important, but some objectives are more important than others.*

Taking such situation into consideration, we propose in this paper the Progressive Addition of Objectives (PAO) heuristic. PAO is a greedy method that starts with a reduced subset of objectives, in particular, those identified as primary objectives. Those objectives are optimized until convergence/stagnation is detected. Subsequently, an additional objective is selected among the remaining ones, by identifying the one that yields a higher diversity of the primary objectives. PAO applies a performance indicator to establish what objective are better candidates to be enrolled in the optimization as it proceeds.

The rest of this paper is organized as follows. Section 2 presents the formal background necessary to introduce the PAO proposal. Subsequently, in Section 3 special attention is given to VorEAl, as it will be extended as part of the paper. Section 4 illustrates the discussion regarding the need of the PAO heuristic by presenting an extension to VorEAl and the corresponding improved results. After that, Section 5 introduces two PAO variants. This is followed by an experimental study that is presented in Section 6. Finally, Section 7 elaborates on the results and discusses possible directions for future research.

2 FOUNDATIONS

As stated, a MOP is an optimization problem where a set of objective functions $f_1(\boldsymbol{x}),\ldots,f_M(\boldsymbol{x})$ should be jointly optimized; formally,

$$\min\ \boldsymbol{F}(\boldsymbol{x}) = \langle\, f_1(\boldsymbol{x}),\ldots,f_M(\boldsymbol{x})\,\rangle\,;\ \boldsymbol{x}\in\mathcal{S}\,; \qquad (1)$$

where $\mathcal{S}\subseteq\mathcal{D}$ is known as the feasible set and could be expressed as a set of restrictions over the decision or search space $\mathcal{D}$. The image set $\mathcal{O}\subseteq\mathbb{R}^M$ of $\mathcal{S}$ produced by the vector-valued function $\boldsymbol{F}(\cdot)$ is called feasible objective set or criterion set.

The solution to this type of problem is a set of trade-off points. The optimality of a solution can be expressed in terms of the Pareto dominance relation.

Definition 2.1 (Pareto dominance). In the optimization problem (1) and having $\boldsymbol{x},\boldsymbol{y}\in\mathcal{S}$, $\boldsymbol{x}$ is said to dominate $\boldsymbol{y}$ (expressed as $\boldsymbol{x}\prec\boldsymbol{y}$) if $\forall j=1,\ldots,M$: $f_j(\boldsymbol{x})\leq f_j(\boldsymbol{y})$ and $\exists i\in\{1,\ldots,M\}$: $f_i(\boldsymbol{x})<f_i(\boldsymbol{y})$. The *non-dominated subset* $\mathcal{A}^*$ of set $\mathcal{A}\subseteq\mathcal{S}$ is defined as

$$\mathcal{A}^* = \left\{\boldsymbol{x}\in\mathcal{A}\,\middle|\,\nexists\boldsymbol{x}'\in\mathcal{A}:\boldsymbol{x}'\prec\boldsymbol{x}\right\}\,. \qquad (2)$$

The solution of (1) is $\mathcal{S}^*$, the non-dominated subset of $\mathcal{S}$. $\mathcal{S}^*$ is known as the *efficient set* or *Pareto-optimal set* [6]. Its image in objective space is known as the *Pareto-optimal front*, $\mathcal{O}^*$.

As finding the explicit formulation of $\mathcal{S}^*$ is often impossible, generally, an algorithm solving (1) yields a discrete non-dominated set, $\mathcal{P}^*$, that approximates $\mathcal{S}^*$. The image of $\mathcal{P}^*$ in objective set, $\mathcal{PF}^*$, is known as the *non-dominated front*.

2.1 Quality of Solutions

The crucial task is how to measure the performance in the multi-objective setting, i.e. how to asses the relation of $\mathcal{PF}^*$ to $\mathcal{O}^*$. Several performance indicators have been proposed including the hypervolume indicator or the R indicators (see [28, 29] for an overview). Each indicator concentrates on special desired characteristics of the front approximation while one frequently discussed aim is that elements of $\mathcal{PF}^*$ should be evenly spread along the true Pareto front in order to present an unbiased solution set to the decision maker.

The hypervolume indicator, $I_{\text{hyp}}(\mathcal{A})$, [1] computes the volume of the region, H, delimited by a given set of points, $\mathcal{A}$, and a set of reference points, $\mathcal{N}$.

$$I_{\text{hyp}}\left(\mathcal{A}\right) = \text{volume}\left(\bigcup_{\forall\boldsymbol{x}\in\mathcal{A};\forall\boldsymbol{y}\in\mathcal{N}}\text{hypercube}(\boldsymbol{x},\boldsymbol{y})\right)\,. \qquad (3)$$

In order to compare different solutions the indicator should be transformed into a relative formulation, as proposed by the binary hypervolume indicator,

$$I_{\text{hyp}}\left(\mathcal{A},\mathcal{B}\right) = I_{\text{hyp}}\left(\mathcal{A}\right) - I_{\text{hyp}}\left(\mathcal{B}\right)\,. \qquad (4)$$

Epsilon indicators [16] are a set of performance indicators that rely on the epsilon dominance concept. This indicator was proposed to measure how close the current non-dominated solution individuals front, $\mathcal{PF}^*_t$, is to the Pareto-optimal front, $\mathcal{O}^*$.

Additive epsilon dominance is defined as:

Definition 2.2 (Additive ε-Dominance Relation). For the optimization problem specified in (1) and having $\boldsymbol{x}_1,\boldsymbol{x}_2\in\mathcal{D}$, $\boldsymbol{x}_1$ is said to additively ε-dominate $\boldsymbol{x}_2$ (expressed as $\boldsymbol{x}_1\preccurlyeq_{\epsilon+}\boldsymbol{x}_2$) iff $f_j(\boldsymbol{x}_1)\leq\varepsilon+f_j(\boldsymbol{x}_2)$.

The additive epsilon indicator, $I_{\varepsilon+}$, is a relative indicator that expresses the minimum value of ϵ that is necessary to make a set $\mathcal{A}$ ε-dominate a set $\mathcal{B}$, that is,

$$I_{\varepsilon+}(\mathcal{A},\mathcal{B}) = \inf_{\varepsilon\in\mathbb{R}} \{\forall \boldsymbol{y} \in \mathcal{B},\ \exists \boldsymbol{x} \in \mathcal{A} \text{ such that } \boldsymbol{x} \preccurlyeq_{\epsilon+} \boldsymbol{y}\}\ . \quad (5)$$

2.2 EMOAs for Many-Objective Optimization

A recent generation of EMOAs exploits existing performance indicators for their selection processes. The $\mathcal{S}$-metric selection evolutionary multiobjective optimization algorithm (SMS-EMOA) [4] belongs to that group of approaches. SMS-EMOA is a steady-state algorithm. Which means that, in every iteration, only one individual is created and only one has to be deleted from the population in each generation. The hypervolume is not computed exactly. Instead, the k-greedy strategy is employed. These decisions were made in the hope of tackling the high computational demands of computing the hypervolume.

The key element of SMS-EMOA is the method for determining which element of the population will be substituted by the offspring. This is done, by applying a non-domination ranking. From the individuals that are dominated by the rest of the population, one individual is selected such that it has the minimum contribution to the hypervolume of the set. This individual is to be removed from the population and substituted by a new individual generated by the usual variation operators. It may happen, that there is only one non-dominated front (all individuals are non-dominated). In this case, the individual with least hypervolume contribution is selected.

Another promising line comes from the reference-point-based many-objective version of the nondominated sorting genetic algorithm (NSGA-II), denominated NSGA-III [11]. Similar to NSGA-II, NSGA-III employs the Pareto non-dominated sorting to partition the population into a number of fronts. In the last front, instead of using the crowding distance to determine the selected solutions a novel niche-preservation operator is applied. This niche-preservation operator relies on reference points organized in a hyperplane to promote diversification of the population. Therefore, solutions associated with less crowded reference points are more likely to be selected. A sophisticated normalization is incorporated into NSGA-III to effectively handle objective functions of different scales.

2.3 Detecting Convergence in EMOAs

The formal determination of convergence or optimality criteria in MOPs (and EMOAs, for that matter) is often impossible when gradient information is not available. This is a common situation in real-world applications. Because of that, sophisticated heuristic stopping criteria have become a subject of intensive research [19, 26].

The on-line convergence detection criterion (OCD) [27] is a robust method for convergence detection. OCD computes a set of performance indicators applying them to a given number of consecutive populations. Relying on the values of the performance indicators, OCD determines if they have remained stable in a non-progress state applying a statistical hypothesis test. In this work, we use OCD to determine if the algorithm is stagnating and, therefore, it is time to add more secondary objectives.

3 THE VORONOI DIAGRAM EVOLUTIONARY ALGORITHM

As stated in the introduction, anomaly detection can be posed as a particular case of classification problems. Here data items must be tagged either as 'normal' or 'anomalous'. That is, relying on a dataset $\Psi = \left\{\boldsymbol{x}^{(i)}, y^{(i)}\right\}$, where, without loss of generality, we can state that $\boldsymbol{x} \in \mathbb{R}^n$ and $y^{(i)} \in \{\text{normal}; \text{anomaly}\}$ obtain a classifier, $M(\boldsymbol{x};\boldsymbol{\phi}) \rightarrow \{\text{normal}; \text{anomaly}\}$, that correctly detects instances that correspond to each of the two categories. Because of this fact, the existing metrics devised to assess the quality of a classification algorithm are also applicable in this context. For this particular problem, the most relevant metrics are accuracy, recall and specificity, although many more could also be of use. These metrics rely on the number of true positives, t_{p}, false positives, f_{p}, false negatives, f_{n}, and true negatives t_{n} produced by a given model M. Accuracy measures the proportion of true results (both true positives and true negatives) regard to the total number of elements in the dataset by computing $a(M) = (t_{\text{p}}+t_{\text{n}})/(t_{\text{p}}+f_{\text{p}}+f_{\text{n}}+t_{\text{n}})$. On the other hand, recall gauges the ratio between the true positives and false negatives as $r(M) = t_{\text{p}}/(t_{\text{p}}+f_{\text{n}})$.

Voronoi diagrams are geometrical constructs that partition a given space and can be used for classification. Any set of points, known as *Voronoi sites*, in a given n-dimensional Euclidean space $\mathcal{E}$ defines a *Voronoi diagram*, i.e., a partition of that space into *Voronoi cells*: the cell corresponding to a given site S is the set of points whose closest site is S. The boundaries between Voronoi cells are the medians of the $[S_i S_j]$ segments, for neighbor Voronoi sites S_i and S_j. Though originally defined in two or three dimensions, there exist several algorithmic procedures to efficiently compute Voronoi diagrams in any dimension.

Voronoi diagrams offer a compact classification representation by attaching to each Voronoi cell (or, equivalently, to the corresponding Voronoi site), a Boolean label. The resulting Voronoi diagram is a partition of the space into 2 subsets: the 'normal' cells are the shape/volume, and the 'anomalous' cells are the outside of the shape/volume.

This representation allows Voronoi diagrams to be evolved in order to use them for anomaly detection. In this case, the genotype is a (variable length) list of labeled Voronoi sites, and the phenotype is the corresponding partition in the space into two subsets. More generally, any piece-wise constant function on the underlying space can be represented by a similar representation by using real-valued labels.

When dealing with anomalies, the dataset is generally highly imbalanced, as, usually, there are fewer 'anomalous' instances than 'normal' ones. If only the classification accuracy is used, the error contribution of the anomalies will be reduced and hence the model will be biased to not regard them. Furthermore, as already mentioned, the anomaly detection problem requires that the classifier is not only able to correctly classify the 'normal' and 'anomalous' instances present in the training dataset but is also capable of detecting when a given input falls in an area that was not covered by data of the training set and, therefore, also can be interpreted as an anomaly.

Consequently, every individual represents a Voronoi diagram as a set of sites, $\mathcal{I} = \{S_i\}$, where each site has an associated label,

$\boldsymbol{S}.\ell \in \{\text{normal}, \text{anomaly}\}$. Relying on that, that individual can be used as a classifier as

$$\text{clfy}(\mathcal{I}, \boldsymbol{x}) = \boldsymbol{S}^*.\ell \text{ with } \boldsymbol{S}^* = \arg\min_{\boldsymbol{S}_i \in \mathcal{I}} \|\boldsymbol{x} - \boldsymbol{S}_i\| . \quad (6)$$

It is possible to prompt the Voronoi diagrams (individuals) to represent the known data in a form as compact as possible by expressing that as the relation between the volumes of the Voronoi cell and the convex hull of the training data that it contains. Let $\mathcal{I} = \{S_i, i = 1 \dots n_{\mathcal{I}}\}$ be a Voronoi diagram, and, for each cell C_i, let $v_i \in \mathbb{R}^+$ be its volume and $\mathcal{D}_i$ the set of data points it contains, i.e., $\mathcal{D}_i = \left\{\boldsymbol{x} \in \Psi; d(x, S_i) \leq d(x, S_j) \forall i \neq j\right\}$, d being the n-dimensional Euclidean distance. We can then define the individual compactness as the sum, for each cell, of the ratio of the volume of the convex hulls of $\mathcal{D}_i$ and the volume of the cell,

$$c(\mathcal{I}) = \begin{cases} \sum_i (|\mathcal{D}_i| - n) \dfrac{\text{volume}(\text{convex\_hull}(\mathcal{D}_i))}{v_i} & \text{if } |\mathcal{D}_i| > n\,, \\ 0 & \text{in other case}\,. \end{cases} \quad (7)$$

Maximizing compactness will produce cells that contain the data in a form as tight as possible. However, the compactness objective can be complemented by one that promotes the existence of empty cells that represent areas of the input domain that are now present in the training data. Such objective would take care of sites with small $\mathcal{D}_i$'s and promote that they become empty as the evolution takes place. A form of representing this is by computing the total volume of cells with an anomaly label of an individual and rate it by the number of elements it contains,

$$v(\mathcal{I}) = \sum_{i, S_i.\ell = \text{anomaly}} \frac{v_i}{1 + 2\ln(|\mathcal{D}_i| + 1)} . \quad (8)$$

Following the above presentation, the problem of finding Voronoi diagrams for anomaly detection can be formalized as the many-objective optimization problem

$$\max \boldsymbol{F} = \langle a(\mathcal{I}), r(\mathcal{I}), c(\mathcal{I}), v(\mathcal{I}) \rangle\,;\ \mathcal{I} \in \mathcal{D}\,, \quad (9)$$

where $\mathcal{D}$ is the set of all possible Voronoi diagrams in the problem domain.

Two variation operators have been put forward to operate on Voronoi diagrams. The mutation operator acts on two levels. At an individual level, a new Voronoi site can be added, at a randomly chosen position, with a random label; or a randomly chosen Voronoi site can be removed. At a site level, Voronoi sites can be moved around in the space (the well-known self-adaptive Gaussian mutation has been chosen here, inspired by evolution strategies) or the label of a Voronoi site can be changed.

The crossover operator takes two Voronoi diagrams as argument and respects the locality of the representation. Voronoi sites that are close to each other should have more chance to stay together than Voronoi sites that are far apart. This is achieved by the geometric crossover that operates by creating a random cutting hyperplane, and exchanging the Voronoi sites from both sides of the hyperplane, ensuring that the resulting diagrams meet the minimum length bound, n_{min}. Both operators are described in detail in [20, 21].

VorEAl follows a $(\mu + \lambda)$ scheme where, in every iteration, an offspring population of λ individuals, $\mathcal{P}_{\text{off}}$, is created from the current one, $\mathcal{P}_t$, by applying the variation operators described above. Subsequently, the best μ individuals are kept for the next iteration population $\mathcal{P}_{t+1}$ and the rest are disregarded. In this paper, the selection of SMS-EMOA and of NSGA-III are applied and compared as they have been shown to yield substantially better results when confronted with many-objective optimization problems.

From the final population, $\mathcal{P}_{\text{final}}$, a committee of individuals, $\mathcal{P}^* \subseteq \mathcal{P}_{\text{final}}$, is selected. This committee contains the best ρ-percent of $\mathcal{P}_{\text{final}}$ in terms of their accuracy. $\mathcal{P}^*$ is used to compute predictions using a majority voting classifier approach.

$$\text{clfy}\left(\mathcal{P}^*, \boldsymbol{x}\right) = \arg\max_{\ell \in \{\text{normal}, \text{anomaly}\}} \left|\left\{\text{clfy}(\mathcal{I}, \boldsymbol{x}) = \ell; \mathcal{I} \in \mathcal{P}^*\right\}\right| . \quad (10)$$

4 IMPACT OF ADDING A LENGTH MINIMIZATION OBJECTIVE

The proposal put forward by this paper was obtained while enhancing VorEAl by improving the selection mechanism and adding a new Voronoi diagram size minimization objective to VorEAl meant for producing diagrams as with as few sites (or cells) as possible. This is an important feature as the computational complexity of using VorEAl as a predictor is bound to the number of sites.

This new size minimization objective was formulated as

$$l(\mathcal{I}) = \frac{1}{1 + 0.01\left(|\mathcal{I}| - n_{\text{min}}\right)}\,, \quad (11)$$

where n_{min} is the lower bound for individual size. $l()$ has the advantage over the direct use of the size of the individual, $|\mathcal{I}|$, that it is to be maximized, as the rest of the objective functions and bounded in $[0, 1]$.

Consequently, the optimization problem of VorEAl becomes

$$\max \mathcal{F} = \langle a(\mathcal{I}), r(\mathcal{I}), c(\mathcal{I}), v(\mathcal{I}), l(\mathcal{I}) \rangle\,;\ \text{with } \mathcal{I} \in \mathcal{D}\,. \quad (12)$$

The results of evolving this instance of VorEAl produced unexpected results as the introduction of the new objective degraded the performance significantly.

In order to understand this phenomenon, we analyzed the impact of different combinations of objectives. The two-spirals test problem (see Section 6 for details) was addressed with instances of VorEAl identically configured but with different combinations of objectives, starting from just using accuracy and recall and adding one of the remaining three objectives. Figure 1 illustrates the impact of adding the objectives. It is particularly relevant to analyze the three three-objective instances. Here it becomes evident that adding the $l()$ objective makes almost impossible to optimize of the other objectives. In that case, it is noticeable that the accuracy attained remains at a very low value, almost comparable to making a random choice. In the other two three-objective cases, although it is noticeable that the algorithm does not progress as fast as in the two-objective case towards the optimal values of accuracy and recall, this can be attributed to the challenge of coping with an additional objective with the same population size. The remaining VorEAl instances in the figure serve to validate the point made in the previous paragraph.

5 PROGRESSIVE ADDITION OF OBJECTIVES

Taking into account the previous observations and discussions, we elaborated a greedy methodology to progressively select which objectives to add and when to do it. The result of this is the heuristic for progressive addition of objectives (PAO).

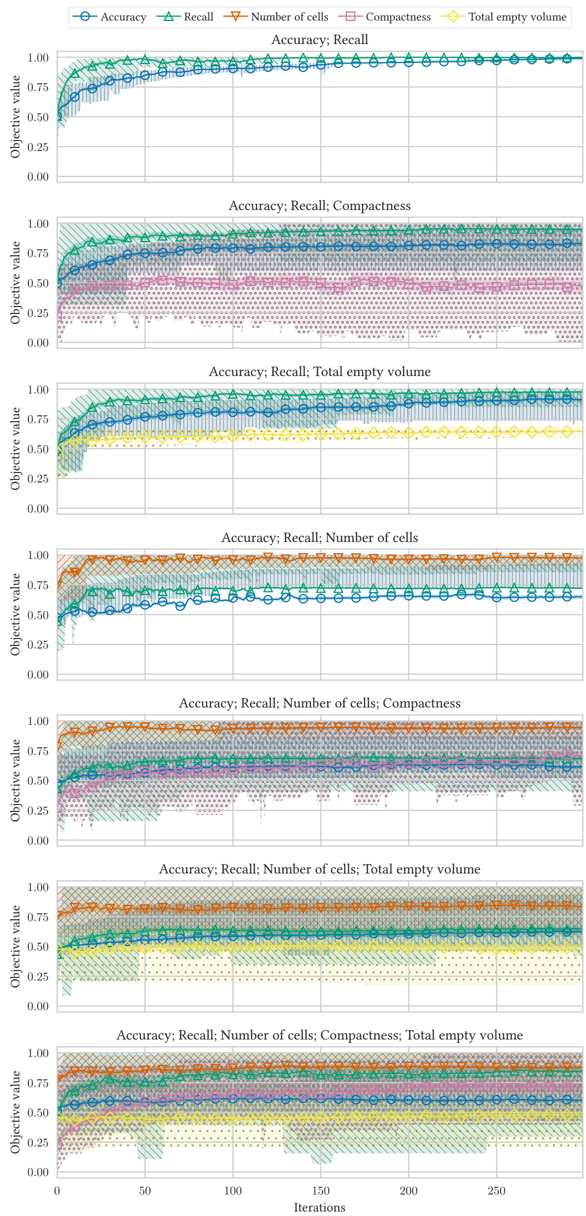


Figure 1: Progression of the objective values of different runs of VorEAl configured with different sets of objectives. Mean, minimum and maximum values of the objective functions are shown. Values of the compactness objective have been scaled to the $[0, 1]$ interval.

In order to describe PAO, is it convenient to introduce the following notation:

- VorEAl$(\mathcal{P}, \mathcal{F}^*)$: An instance of VorEAl with population $\mathcal{P}$ and $\mathcal{F}^* \subseteq \mathcal{F}$ a subset of objective functions $\mathcal{F}$.
- $\texttt{evolve}(v, \Delta t)$: A function that evolves a VorEAl instance v until convergence is detected by the OCD method. It returns the populations $\mathcal{P}_t$ and $\mathcal{P}_{t-\Delta t}$ that one corresponding to the last population and the other to the one obtained Δt iterations before.
- $\texttt{evolve\_t}_{\max}(V, \Delta t, t_{\max})$: A function that evolves a VorEAl instance v for $t_{\max}$ iterations. It returns the population $\mathcal{P}_t$ corresponding to the last iteration.

PAO starts with a set of primary objectives, $\mathcal{F}^{\text{prim}} \subseteq \mathcal{F}$, and a set with the remaining (secondary) ones, $\mathcal{F}^{\text{sec}} \subseteq \mathcal{F}$ such that $\mathcal{F}^{\text{prim}} \cap \mathcal{F}^{\text{sec}} = \varnothing$ and $\mathcal{F}^{\text{prim}} \cup \mathcal{F}^{\text{sec}} = \mathcal{F}$. $\mathcal{F}^{\text{prim}}$ and $\mathcal{F}^{\text{sec}}$ must be provided as input. They would probably be best provided by an expert of decision maker.

Initially, an instance $v^* = \text{VorEAl}(\mathcal{P}_0, \mathcal{F}^{\text{prim}})$ is created with a random initial population $\mathcal{P}_0$. Function $\texttt{evolve}(\text{VorEAl}(v^*, \Delta t)$ is invoked. After function evolve() finishes we have the population $\mathcal{P}_t$ at time t for the instance v^* and, $\mathcal{P}_{t-\Delta t}$ the population at Δt iterations before. In this condition an objective $f \in \mathcal{F}^{\text{sec}}$ is selected and moved to $\mathcal{F}^{\text{prim}}$,

$$\mathcal{F}^{\text{prim}} = \mathcal{F}^{\text{prim}} \cup \{f\}\,;\ \mathcal{F}^{\text{sec}} = \mathcal{F}^{\text{sec}} \setminus \{f\}\,. \tag{13}$$

The VorEAl instance is extended with the new set of primary objectives and the population obtained so far $v^* = \text{VorEAl}(\mathcal{P}_t, \mathcal{F}^{\text{prim}})$. The process is repeated until $\mathcal{F}^{\text{sec}}$ is empty.

At this point, the main issue is what procedure to follow to identify what objectives should be added. There is a rather large body of work on how to detect related objectives mostly derived from the works on objective reduction [7]. However, those works, although somewhat related, are directed in the opposite direction.

The procedure to follow is to create one VorEAl instance for every objective in $\mathcal{F}^{\text{sec}}$ such that,

$$\boldsymbol{V} = \left\{v_i := \text{VorEAl}(\mathcal{P}_{t-\Delta t}, \mathcal{F}^{\text{prim}} \cup \{f_i\}); \forall f_i \in \mathcal{F}^{\text{sec}}\right\}\,, \tag{14}$$

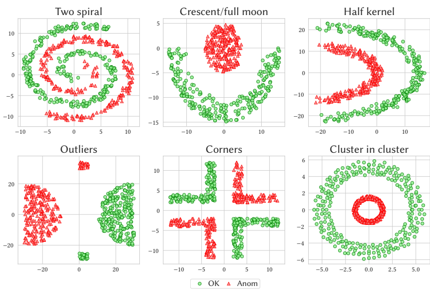
Each of these instances would evolve the population $\mathcal{P}_{t-\Delta t}$ for Δt iterations by invoking $\texttt{evolve\_t}_{\max}()$ on each instance to produce a set of populations,

$$\mathfrak{P}^+ = \left\{\mathcal{P}_i^+\right\} = \{\texttt{evolve\_t}_{\max}(v_i, \Delta t); \forall v_i \in \mathcal{F}^{\text{sec}}\}\,. \tag{15}$$

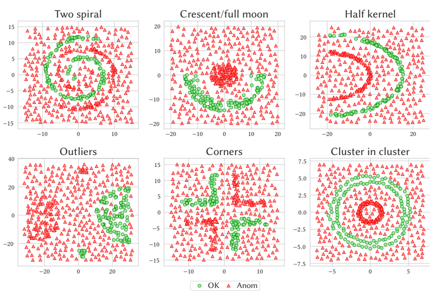
Every the population in $\mathfrak{P}^+$ is expected to capture the impact of its corresponding objective on the evolutionary process. Following the preliminary discussion, the idea here would be to select the objective that degrades as little as possible the convergence and diversity of the population. This is achieved by comparing these populations with the $\mathcal{P}_t$ previously obtained and selecting the objective of $\mathcal{F}^{\text{sec}}$ that had the least impact on the values of $\mathcal{F}^{\text{prim}}$. That is, having a comparison function $\delta_{\mathcal{F}}(\mathcal{A}, \mathcal{B})$ that determines how similar are $\mathcal{A}$ and $\mathcal{B}$ in terms of the objective functions in $\mathcal{F}$, we can formulate this selection process as

$$f_i = \arg\min \delta_{\mathcal{F}^{\text{prim}}}\left(\mathcal{P}_i^+, \mathcal{P}_t\right)\,. \tag{16}$$

Function $\delta()$ can be defined relying on the performance indicators introduced in Section 2. Consequently, we will denominate the PAO variant that applied the hypervolume indicator as $\mathcal{S}$-PAO and the one using the ε indicator as ε-PAO.



(a) Training datasets.



(b) Test datasets.

Figure 2: Training and testing datasets. Test set anomalies present in the test datasets are generated using the procedure described in Section 6.

6 EXPERIMENTS

The previous discussion and proposal must be complemented by a set of experiments that establish their validity. The primary objective of the experiments is to assert if the PAO heuristic does manage to speed up the optimization process by preserving population diversity. It is also important to verify which of the two PAO variants produces the best results. Similarly, we will also be assessing which of the two selection methods performs best. For that reason, different combinations of PAO and selection methods were tested together with the regular VorEAl with no objective addition heuristic. VorEAl parameters were kept as in [20], a $\Delta t = 30$ was used and algorithms were left to run for 1500 iterations.

Experiments involved six classification benchmarks problems: the 'two spiral', 'crescent and full moon', 'half densities', 'corners', 'outliers' and 'cluster in cluster' problems. They have the advantage that they can be visualized in 2D while still posing a substantial challenge to the algorithm. One key element that must be addressed is the ability of the method to detect anomalies that were present in the original dataset as well as those that were not present. Six tests were prepared with that goal in mind by adding random anomaly data in the areas that did not have any data in the training dataset. The resulting training and test datasets can be observed in Figure 2.

Other methods were included in the experiments in order to provide grounds for comparison with similar approaches as well as well-known approaches. In particular, we included the negative selection algorithm (NSA) [15] using both variable-sized hyperspheres and hyper-rectangles. For fair comparisons, we applied the NSA^{+}_{sp} and NSA^{+}_{re} in which non-self training samples are subsequently used to enrich the detector library generated by NSA.

Similarly, we have included in the experiments two well-known classifiers: one-class vector machines (SVMs) [23] and the naive Bayes classifier [12]. Support vector machines are particularly capable of yielding adequate results, while the naive Bayes classifier would serve as a baseline.

The stochastic nature of the algorithms being analyzed calls for the use of an experimental methodology that relies on statistical hypothesis tests. Using those tests, we are able to determine in a statistically sound way if one algorithm instance outperforms another. The topic of assessing stochastic classification algorithms is studied in depth in [14]. There, it is shown that the Bergmann–Hommel [3] procedure is the most suitable for our class of problem.

In all cases, we have used a base level of significance of 0.05 and we ran all experiment instances 50 times. The results of these experiments are shown as box plots in Figure 3. The results indicate that $\mathcal{S}$-PAO heuristic managed to yield the best results. Similarly, it is also important to note that the NSGA-III selection outperforms SMS-EMOA. This is probably because it manages to maintain diversity.

As many tests were carried out, a comprehensive analysis of the results is rather difficult as it implies cross-examining and comparing the results presented separately. Consequently, we present them in summarized form in Table 1. This validates the previous discussion from a statistical point of view.

To further simplify the understanding of the results, we decided to adopt a more integrative representation like the one proposed in [2]. This representation groups either by problem or by classification metric the results of the algorithms. It does so by computing the number of times a given algorithm was better than the others. Figure 4 conveys these analyses and further establishes how the combination of $\mathcal{S}$-PAO and NSGA-III selection yielded the best results.

7 CONCLUSIONS

In this paper, we have introduced the PAO heuristic, meant to deal with the severe loss of diversity that EMOAs may face when applied on objective functions that have different degrees of complexity. PAO emerged from the need to extend VorEAl and is inspired by its use in real-world anomaly detection problems. Consequently, we have carried out a proof-of-concept study, meant to test its working hypothesis. The results are of relevance, as the introduction of the PAO heuristic managed to improve the performance of the modified VorEAl when confronted with the test sets.

However, work in this direction is only starting. The precepts put forward in this paper are not limited to the context of VorEAl, or anomaly detection, for that matter. The PAO heuristics can be applied in any many-objective problem with objectives of different degrees of complexity and with the potential of dragging the population at a very fast rate to sub-optimal areas of the search space. One of such application scenarios is controlling bloat in genetic programming.

Similarly, connections between this proposal and other topics like bi-level optimization, objective reduction and performance indicators, should be properly dealt with. Finally, it is also necessary

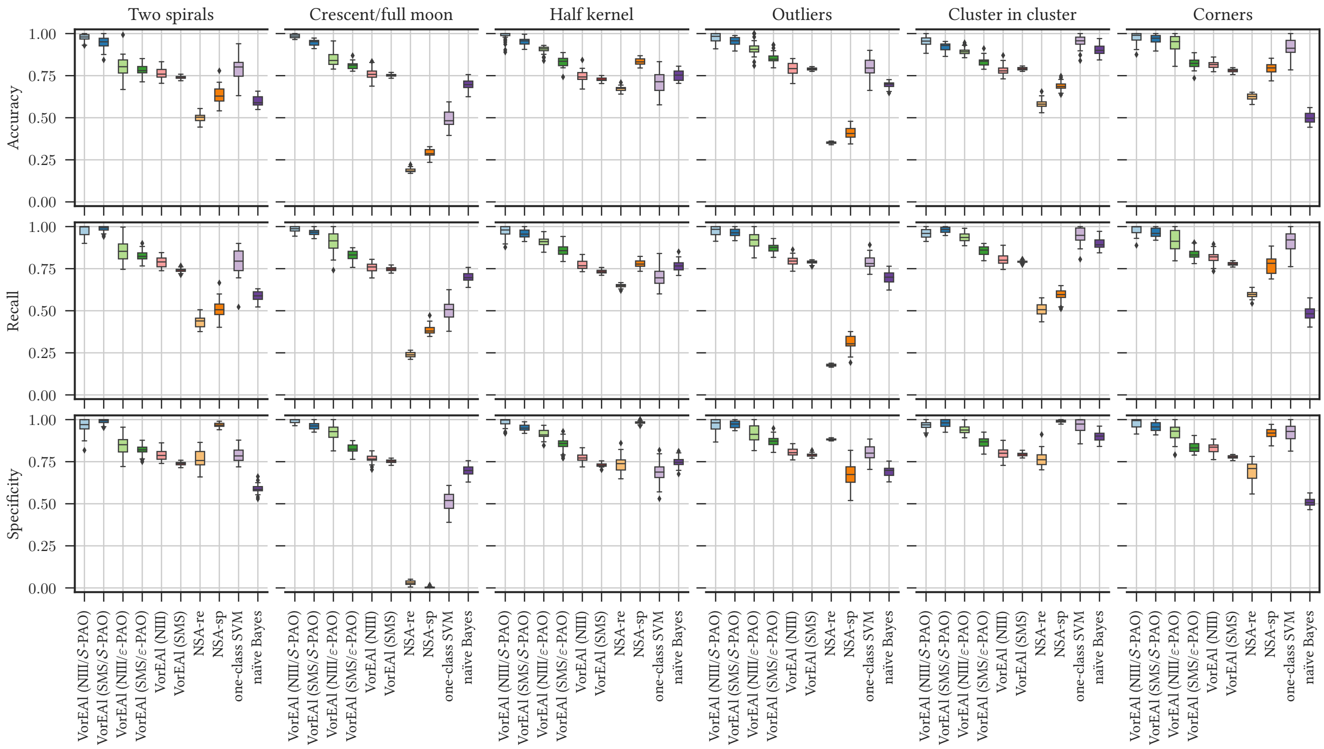


Figure 3: Box plots of the different combinations of $\mathcal{S}$-PAO and ε-PAO heuristics and selection operators based on NSGA-III (NIII) and SMS-EMOA (SMS), VorEAl without using the PAO heuristics and the other methods included in the comparison.

Table 1: Summarized results of the Bergmann–Hommel statistical hypothesis tests on the two spirals, crescent/full moon, half kernel, outliers, corners and cluster in cluster data sets. Legend: +, row better than column; ~, homogeneous results and −, row worst than column.

Two spirals

		VorEAl (NIII/$\mathcal{S}$-PAO)	VorEAl (SMS/$\mathcal{S}$-PAO)	VorEAl (NIII/ε-PAO)	VorEAl (SMS/ε-PAO)	VorEAl (NIII)	VorEAl (SMS)	NSA-re	NSA-sp	one-class SVM	naive Bayes
Accuracy	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	~	+	+	+	+	~	+
	VorEAl (SMS/ε-PAO)	−	−	~	·	+	+	+	+	~	+
	VorEAl (NIII)	−	−	−	−	·	+	+	+	−	+
	VorEAl (SMS)	−	−	−	−	−	·	+	+	−	+
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	−	−	+	·	−	+
	one-class SVM	−	−	~	~	+	+	+	+	·	+
	naive Bayes	−	−	−	−	−	−	+	−	−	·
Recall	VorEAl (NIII/$\mathcal{S}$-PAO)	·	~	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	~	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	+	+
	VorEAl (NIII)	−	−	−	−	·	+	+	+	~	+
	VorEAl (SMS)	−	−	−	−	−	·	+	+	−	+
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	−	−	+	·	−	−
	one-class SVM	−	−	−	−	~	+	+	+	·	+
	naive Bayes	−	−	−	−	−	−	+	+	−	·
Specificity	VorEAl (NIII/$\mathcal{S}$-PAO)	·	−	+	+	+	+	+	~	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	+	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	−	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	−	+	+
	VorEAl (NIII)	−	−	−	−	·	+	+	−	~	+
	VorEAl (SMS)	−	−	−	−	−	·	−	−	−	+
	NSA-re	−	−	−	−	−	+	·	−	−	+
	NSA-sp	~	−	+	+	+	+	+	·	+	+
	one-class SVM	−	−	−	−	~	+	+	−	·	+
	naive Bayes	−	−	−	−	−	−	−	−	−	·

Cresc./full moon

		VorEAl (NIII/$\mathcal{S}$-PAO)	VorEAl (SMS/$\mathcal{S}$-PAO)	VorEAl (NIII/ε-PAO)	VorEAl (SMS/ε-PAO)	VorEAl (NIII)	VorEAl (SMS)	NSA-re	NSA-sp	one-class SVM	naive Bayes
Accuracy	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	+	+
	VorEAl (NIII)	−	−	−	−	·	~	+	+	+	+
	VorEAl (SMS)	−	−	−	−	~	·	+	+	+	+
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	−	−	+	·	−	−
	one-class SVM	−	−	−	−	−	−	+	+	·	−
	naive Bayes	−	−	−	−	−	−	+	+	+	·
Recall	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	+	+
	VorEAl (NIII)	−	−	−	−	·	+	+	+	+	+
	VorEAl (SMS)	−	−	−	−	−	·	+	+	+	+
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	−	−	+	·	−	−
	one-class SVM	−	−	−	−	−	−	+	+	·	−
	naive Bayes	−	−	−	−	−	−	+	+	+	·
Specificity	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	+	+
	VorEAl (NIII)	−	−	−	−	·	+	+	+	+	+
	VorEAl (SMS)	−	−	−	−	−	·	+	+	+	+
	NSA-re	−	−	−	−	−	−	·	+	−	−
	NSA-sp	−	−	−	−	−	−	−	·	−	−
	one-class SVM	−	−	−	−	−	−	+	+	·	−
	naive Bayes	−	−	−	−	−	−	+	+	+	·

Half kernel

		VorEAl (NIII/$\mathcal{S}$-PAO)	VorEAl (SMS/$\mathcal{S}$-PAO)	VorEAl (NIII/ε-PAO)	VorEAl (SMS/ε-PAO)	VorEAl (NIII)	VorEAl (SMS)	NSA-re	NSA-sp	one-class SVM	naive Bayes
Accuracy	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	~	+	+
	VorEAl (NIII)	−	−	−	−	·	+	+	−	+	~
	VorEAl (SMS)	−	−	−	−	−	·	+	−	~	−
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	~	+	+	+	·	+	+
	one-class SVM	−	−	−	−	−	~	+	−	·	−
	naive Bayes	−	−	−	−	~	+	+	−	+	·
Recall	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	+	+
	VorEAl (NIII)	−	−	−	−	·	+	+	~	+	~
	VorEAl (SMS)	−	−	−	−	−	·	+	−	+	−
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	~	+	+	·	+	+
	one-class SVM	−	−	−	−	−	−	+	−	·	−
	naive Bayes	−	−	−	−	~	+	+	−	+	·
Specificity	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	~	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	−	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	−	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	−	+	+
	VorEAl (NIII)	−	−	−	−	·	+	+	−	+	+
	VorEAl (SMS)	−	−	−	−	−	·	~	−	+	−
	NSA-re	−	−	−	−	−	~	·	−	+	~
	NSA-sp	~	+	+	+	+	+	+	·	+	+
	one-class SVM	−	−	−	−	−	−	−	−	·	−
	naive Bayes	−	−	−	−	−	+	~	−	+	·

Outliers

		VorEAl (NIII/$\mathcal{S}$-PAO)	VorEAl (SMS/$\mathcal{S}$-PAO)	VorEAl (NIII/ε-PAO)	VorEAl (SMS/ε-PAO)	VorEAl (NIII)	VorEAl (SMS)	NSA-re	NSA-sp	one-class SVM	naive Bayes
Accuracy	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	+	+
	VorEAl (NIII)	−	−	−	−	·	~	+	+	~	+
	VorEAl (SMS)	−	−	−	−	~	·	+	+	~	+
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	−	−	+	·	−	−
	one-class SVM	−	−	−	−	~	~	+	+	·	+
	naive Bayes	−	−	−	−	−	−	+	+	−	·
Recall	VorEAl (NIII/$\mathcal{S}$-PAO)	·	~	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	~	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	+	+
	VorEAl (NIII)	−	−	−	−	·	~	+	+	~	+
	VorEAl (SMS)	−	−	−	−	~	·	+	+	~	+
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	−	−	+	·	−	−
	one-class SVM	−	−	−	−	~	~	+	+	·	+
	naive Bayes	−	−	−	−	−	−	+	+	−	·
Specificity	VorEAl (NIII/$\mathcal{S}$-PAO)	·	~	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	~	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	+	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	−	+	+	+
	VorEAl (NIII)	−	−	−	−	·	+	−	+	~	+
	VorEAl (SMS)	−	−	−	−	−	·	−	+	~	+
	NSA-re	−	−	−	+	+	+	·	+	+	+
	NSA-sp	−	−	−	−	−	−	−	·	−	~
	one-class SVM	−	−	−	−	~	~	−	+	·	+
	naive Bayes	−	−	−	−	−	−	−	~	−	·

Cluster in cluster

		VorEAl (NIII/$\mathcal{S}$-PAO)	VorEAl (SMS/$\mathcal{S}$-PAO)	VorEAl (NIII/ε-PAO)	VorEAl (SMS/ε-PAO)	VorEAl (NIII)	VorEAl (SMS)	NSA-re	NSA-sp	one-class SVM	naive Bayes
Accuracy	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	~	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	−	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	−	~
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	−	−
	VorEAl (NIII)	−	−	−	−	·	−	+	+	−	−
	VorEAl (SMS)	−	−	−	−	+	·	+	+	−	−
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	−	−	+	·	−	−
	one-class SVM	~	+	+	+	+	+	+	+	·	+
	naive Bayes	−	−	~	+	+	+	+	+	−	·
Recall	VorEAl (NIII/$\mathcal{S}$-PAO)	·	−	+	+	+	+	+	+	~	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	+	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	~	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	−	−
	VorEAl (NIII)	−	−	−	−	·	~	+	+	−	−
	VorEAl (SMS)	−	−	−	−	~	·	+	+	−	−
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	−	−	−	−	−	−	+	·	−	−
	one-class SVM	~	−	~	+	+	+	+	+	·	+
	naive Bayes	−	−	−	+	+	+	+	+	−	·
Specificity	VorEAl (NIII/$\mathcal{S}$-PAO)	·	−	+	+	+	+	+	−	~	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	+	·	+	+	+	+	+	~	~	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	−	−	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	−	−	−
	VorEAl (NIII)	−	−	−	−	·	~	+	−	−	−
	VorEAl (SMS)	−	−	−	−	~	·	+	−	−	−
	NSA-re	−	−	−	−	−	−	·	−	−	−
	NSA-sp	+	~	+	+	+	+	+	·	+	+
	one-class SVM	~	~	+	+	+	+	+	−	·	+
	naive Bayes	−	−	−	+	+	+	+	−	−	·

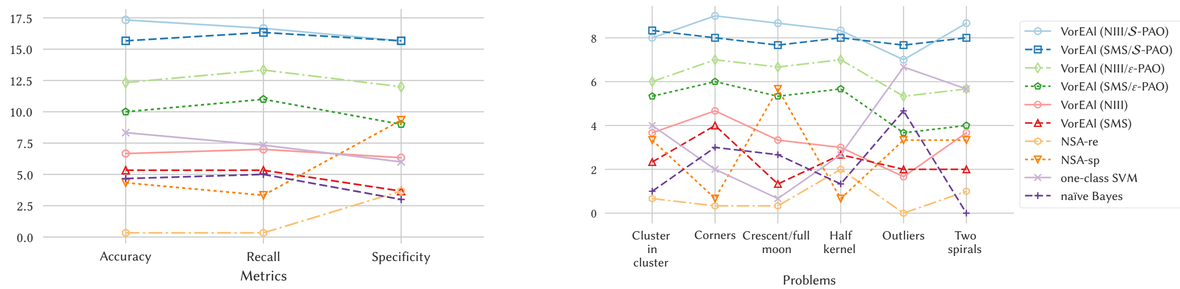
Corners

		VorEAl (NIII/$\mathcal{S}$-PAO)	VorEAl (SMS/$\mathcal{S}$-PAO)	VorEAl (NIII/ε-PAO)	VorEAl (SMS/ε-PAO)	VorEAl (NIII)	VorEAl (SMS)	NSA-re	NSA-sp	one-class SVM	naive Bayes
Accuracy	VorEAl (NIII/$\mathcal{S}$-PAO)	·	~	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	~	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	~	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	~	+	+	+	−	+
	VorEAl (NIII)	−	−	−	~	·	+	+	+	−	+
	VorEAl (SMS)	−	−	−	−	−	·	+	−	−	+
	NSA-re	−	−	−	−	−	−	·	−	−	+
	NSA-sp	−	−	−	−	−	+	+	·	−	+
	one-class SVM	−	−	~	+	+	+	+	+	·	+
	naive Bayes	−	−	−	−	−	−	−	−	−	·
Recall	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	+	~	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	+	+	+	+	−	+
	VorEAl (NIII)	−	−	−	−	·	+	+	+	−	+
	VorEAl (SMS)	−	−	−	−	−	·	+	~	−	+
	NSA-re	−	−	−	−	−	−	·	−	−	+
	NSA-sp	−	−	−	−	−	~	+	·	−	+
	one-class SVM	−	−	~	+	+	+	+	+	·	+
	naive Bayes	−	−	−	−	−	−	−	−	−	·
Specificity	VorEAl (NIII/$\mathcal{S}$-PAO)	·	+	+	+	+	+	+	+	+	+
	VorEAl (SMS/$\mathcal{S}$-PAO)	−	·	+	+	+	+	+	+	+	+
	VorEAl (NIII/ε-PAO)	−	−	·	+	+	+	+	~	~	+
	VorEAl (SMS/ε-PAO)	−	−	−	·	~	+	+	−	−	+
	VorEAl (NIII)	−	−	−	~	·	+	+	−	−	+
	VorEAl (SMS)	−	−	−	−	−	·	+	−	−	+
	NSA-re	−	−	−	−	−	−	·	−	−	+
	NSA-sp	−	−	~	+	+	+	+	·	~	+
	one-class SVM	−	−	~	+	+	+	+	~	·	+
	naive Bayes	−	−	−	−	−	−	−	−	−	·

to study the computational footprint of PAO, in particular in the hypervolume variant.

ACKNOWLEDGMENTS

This work has been funded by project PIA-FSN-P3344-146479. Authors wish to thank the reviewers for their fruitful comments.



(a) Summary by metric. **(b) Summary by problem.**

Figure 4: Summaries of the statistical tests by problem and metric. Higher values are better.

REFERENCES

[1] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. 2009. Theory of the Hypervolume Indicator: Optimal μ-distributions and the Choice of the Reference Point. In *Proceedings of the Tenth ACM SIGEVO Workshop on Foundations of Genetic Algorithms (FOGA'09)*. ACM, New York, NY, USA, 87–102. DOI: http://dx.doi.org/10.1145/1527125.1527138

[2] Johannes Bader. 2010. *Hypervolume-Based Search for Multiobjective Optimization: Theory and Methods*. Ph.D. Dissertation. ETH Zurich, Switzerland.

[3] Beate Bergmann and Gerhard Hommel. 1988. Improvements of general multiple test procedures for redundant systems of hypotheses. In *Multiple Hypothesenprüfung/Multiple Hypotheses Testing*. Springer, 100–115.

[4] Nicola Beume, Boris Naujoks, and Michael Emmerich. 2007. SMS–EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research* 181, 3 (September 2007), 1653–1669. http://ideas.repec.org/a/eee/ejores/v181y2007i3p1653-1669.html

[5] S. Bleuler, M. Brack, L. Thiele, and E. Zitzler. 2001. Multiobjective genetic programming: Reducing bloat using SPEA2. In *Proceedings of the 2001 IEEE Congress on Evolutionary Computation*, Vol. 1. IEEE Press, 536–543. DOI: http://dx.doi.org/10.1109/CEC.2001.934438

[6] Jürgen Branke, Kaisa Miettinen, Kalyanmoy Deb, and Roman S lowiǹski (Eds.). 2008. *Multiobjective Optimization*. Lecture Notes in Computer Science, Vol. 5252. Springer–Verlag, Berlin/Heidelberg.

[7] Dimo Brockhoff and Eckart Zitzler. 2009. Objective Reduction in Evolutionary Multiobjective Optimization: Theory and Applications. *Evolutionary Computation* 17, 2 (2009), 135–166. DOI: http://dx.doi.org/10.1162/evco.2009.17.2.135

[8] Varun Chandola, Arindam Banerjee, and Vipin Kumar. 2009. Anomaly detection: A survey. *ACM Computing Surveys (CSUR)* 41, 3 (2009), 15.

[9] Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. 2007. *Evolutionary Algorithms for Solving Multi-Objective Problems* (second ed.). Springer, New York.

[10] Edwin D. de Jong, Richard A. Watson, and Jordan B. Pollack. 2001. Reducing Bloat and Promoting Diversity Using Multi-objective Methods. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation (GECCO'01)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 11–18.

[11] Kalyanmoy Deb and Himanshu Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (aug 2014), 577–601. DOI: http://dx.doi.org/10.1109/TEVC.2013.2281535

[12] Pedro Domingos and Michael Pazzani. 1997. On the Optimality of the Simple Bayesian Classifier Under Zero-One Loss. *Mach. Learn.* 29, 2-3 (Nov. 1997), 103–130. DOI: http://dx.doi.org/10.1023/A:1007413511361

[13] Carlos M. Fonseca and Peter J. Fleming. 1993. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *Proceedings of the 5th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 416–423. http://dl.acm.org/citation.cfm?id=645513.657757

[14] Salvador García and Francisco Herrera. 2008. An Extension on "Statistical Comparisons of Classifiers over Multiple Data Sets" for all Pairwise Comparisons. *Journal of Machine Learning Research* 9 (2008), 2677–2694.

[15] Z Ji and D Dasgupta. 2004. Real-valued negative selection algorithm with variable-sized detectors. *Lect Notes Comput Sc* 3102 (2004), 287–298.

[16] Joshua Knowles, Lothar Thiele, and Eckart Zitzler. 2006. *A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers*. 214. Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland. revised version.

[17] John R Koza. 1992. *Genetic programming: On the programming of computers by means of natural selection*. Vol. 1. MIT press.

[18] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. 2015. Many-Objective Evolutionary Algorithms. *Comput. Surveys* 48, 1 (2015), 1–35. DOI: http://dx.doi.org/10.1145/2792984

[19] Luis Martí. 2011. *Scalable Multi-Objective Optimization*. Ph.D. Dissertation. Departmento de Informática, Universidad Carlos III de Madrid, Colmenarejo, Spain. http://hdl.handle.net/10016/11896

[20] Luis Martí, Arsene Fansi-Tchango, Laurent Navarro, and Marc Schoenauer. 2016. Anomaly Detection with the Voronoi Diagram Evolutionary Algorithm. In *Proceedings of the 14th International Conference Parallel Problem Solving from Nature (PPSN XIV)*, Julia Handl, Emma Hart, R. Peter Lewis, Manuel López-Ibáñez, Gabriela Ochoa, and Ben Paechter (Eds.). Springer International Publishing, Berlin/Heidelberg, 697–706. DOI: http://dx.doi.org/10.1007/978-3-319-45823-6_65

[21] Luis Martí, Arsene Fansi-Tchango, Laurent Navarro, and Marc Schoenauer. 2016. VorAIS: A Multi-Objective Voronoi Diagram-based Artificial Immune System. In *Proceedings of the 2016 Annual Conference on Genetic and Evolutionary Computation (GECCO'16) Companion*. ACM Press, New York, NY, USA, 11–12. DOI: http://dx.doi.org/10.1145/2908961.2909027

[22] George Orwell. 1945. *Animal Farm*. Secker and Warburg, London, UK.

[23] David M. J. Tax and Robert P. W. Duin. 2004. Support vector data description. *Machine learning* 54, 1 (2004), 45–66.

[24] Lothar Thiele, Kaisa Miettinen, Pekka J Korhonen, and Julian Molina. 2009. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary computation* 17, 3 (jan 2009), 411–36. DOI: http://dx.doi.org/10.1162/evco.2009.17.3.411

[25] Tobias Wagner, Heike Trautmann, and Dimo Brockhoff. 2013. *Preference Articulation by Means of the R2 Indicator*. Springer Berlin Heidelberg, Berlin, Heidelberg, 81–95. DOI: http://dx.doi.org/10.1007/978-3-642-37140-0_10

[26] Tobias Wagner, Heike Trautmann, and Luis Martí. 2011. A Taxonomy of Online Stopping Criteria for Multi-Objective Evolutionary Algorithms. In *6th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2011)*, Ricardo H. C. Takahashi, Kalyanmoy Deb, Elizabeth F. Wanner, and Salvatore Greco (Eds.), Vol. 6576. Springer, Berlin/Heidelberg, 16–30. 10.1007/978-3-642-19893-9_2.

[27] Tobias Wagner, Heike Trautmann, and Boris Naujoks. 2009. OCD: Online Convergence Detection for Evolutionary Multi-Objective Algorithms Based on Statistical Testing. In *5th International Conference on Evolutionary Multi-Criterion Optimization (EMO 2009) (Lecture Notes in Computer Science)*, Matthias Ehrgott, Carlos M. Fonseca, Xavier Gandibleux, Jin-Kao Hao, and Marc Sevaux (Eds.), Vol. 5467. Springer, Berlin/Heidelberg, 198–215.

[28] Eckart Zitzler, Marco Laumanns, Lothar Thiele, Carlos M. Fonseca, and Viviane Grunert da Fonseca. 2002. Why Quality Assessment of Multiobjective Optimizers is Difficult. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002)*, W. B. Langdon, E. Cantú-Paz, K. Mathias, R. Roy, D. Davis, R. Poli, K. Balakrishnan, V. Honavar, G. Rudolph, J. Wegener, L. Bull, M.A. Potter, A.C. Schultz, J.F. Miller, E. Burke, and N. Jonoska (Eds.). Morgan Kaufmann Publishers, San Francisco, California, 666–673.

[29] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation* 7, 2 (2003), 117–132.